



TITLE:

代数的局所コホモロジーの計算法 とそれを用いたスタンダード基底 ・グレブナー基底計算について (実 閉体上の幾何と特異点論への応用)

AUTHOR(S):

鍋島, 克輔; 中村, 弥生; 田島, 慎一

CITATION:

鍋島, 克輔 ...[et al]. 代数的局所コホモロジーの計算法とそれを用いたスタンダード基底・グレブナー基底計算について (実閉体上の幾何と特異点論への応用). 数理解析研究所講究録 2011, 1764: 102-125

ISSUE DATE:

2011-09

URL:

<http://hdl.handle.net/2433/171396>

RIGHT:

代数的局所コホモロジーの計算法とそれを用いた スタンダード基底・グレブナー基底計算について

鍋島克輔*

NABESHIMA, KATSUSUKE

徳島大学大学院ソシオ・アーツ・アンド・サイエンス研究部

INSTITUTE OF SOCIO-ARTS AND SCIENCES, THE UNIVERSITY OF TOKUSHIMA

中村弥生†

NAKAMURA, YAYOI

近畿大学理工学部

FACULTY OF SCIENCE AND ENGINEERING, KINKI UNIVERSITY

田島慎一‡

TAJIMA, SHINICHI

筑波大学大学院数理物質科学研究科

GRADUATE SCHOOL OF PURE AND APPLIED SCIENCES, UNIVERSITY OF TSUKUBA

1 はじめに

代数的局所コホモロジー [10] は、可換代数や代数幾何のみならず、[4], [14] あるいは最近の survey [14] に見られるように様々な分野に応用を持つ重要な概念である。一般の場合に比べ一点に台を持つような代数的局所コホモロジー類は、複素領域におけるデルタ関数の高階の偏導関数の一次結合が定める超関数と見做すことが出来、解析学の徒にも理解しやすい概念であり、特異点の解析を行う際にも有効に用いることができる。

本稿では、孤立特異点を持つ超曲面に対しそのヤコビイデアルに付随した代数的局所コホモロジーで注目した孤立特異点に台を持つものを考える。論文 [30] に従い、孤立特異点に付随する代数的局所コホモロジーを求める計算アルゴリズムを紹介し、更に、これら代数的局所コホモロジーを用いることでイデアルのスタンダード基底やグレブナー基底を求めるアルゴリズム、イデアルメンバーシップ問題を解く方法について述べる。

さて、広中 [11] により 1964 年に導入されたスタンダード基底は、計算代数の観点からも盛んに研究された。Mora, Lazard, Gräbe らの研究 [7, 8, 13, 16] により、(零次元とは限らない一般の次元のイデアルに対しても) スタンダード基底を計算する方法が確立した。また、イデアルが零次元である場合は、Macaulay の inverse system の理論に源流をもつ双対性を利用することでスタンダード基底を求めるアルゴリズムが知られている [2, 15, 18]。これに対し本稿で与えるイデアルメンバーシップ問題の解法やスタンダード基底の計算法は、多変数留数に関する Grothendieck 双対性に基づくことで得られたものであり、従来のものとは異なる観点から導出されたものである。

*nabesima@ias.tokushima-u.ac.jp

†yayoi@math.kindai.ac.jp

‡tajima@math.tsukuba.ac.jp

本稿で紹介する代数的局所コホモロジーの計算アルゴリズムは、計算効率を考慮し動的アルゴリズムとして設計してあるが、実質的な計算は線形計算のみからなるアルゴリズムである。その出力は、イデアルに関する豊富な情報を有しており、これらを利用することでイデアルメンバーシップ問題やスタンダード基底計算を瞬時に行うことが出来る。実際、スタンダード基底を求めるのに必要な計算は代数的局所コホモロジーを求める際に実質的に全て済ませている。従って、スタンダード基底の計算は、代数的局所コホモロジーの計算アルゴリズムの出力に対し、いかなる計算もすることなく、極めて単純な処理を施すだけで完了する。また、本稿の方法でグレブナー基底を求める際は代数的局所コホモロジーに対し線形計算を行う必要があるが、多項式環における計算法と異なり、準素イデアル分解を必要としない点も強調しておきたい。既に述べたとおり、本稿で与えるアルゴリズムは、実質的な計算は線形計算のみからなる計算法であり、一般的なスタンダード基底の計算法である Mora の tangent cone アルゴリズム [8, 16, 17] やグレブナー基底を求める Buchberger アルゴリズム [5] で必須な S 多項式の計算を必要としない。また、アルゴリズムの終了判定が容易であり計算効率が良い。

本稿ではまず特異点に付随する代数的局所コホモロジーの計算方法を紹介する。代数的局所コホモロジー計算の先行研究としては [1, 25, 26] があるが、本稿ではこれらのアルゴリズムを改良・効率化した計算方法 [30] を紹介する。また、代数的局所コホモロジーの応用として、スタンダード基底計算、グレブナー基底計算、イデアルメンバーシップ問題について述べる。

本稿の後半部では、定義多項式がパラメータを持つ場合を扱う。一般に、イデアルがパラメータを含む場合、そのスタンダード基底の先頭項 (initial 項) 集合はパラメータに依存して変化し、スタンダード基底の構造そのものがパラメータ値に大きく依存することになる。従って、パラメータ付きスタンダード基底計算では、パラメータ空間の stratification と各 stratum におけるスタンダード基底計算を行うことが必要となる。我々は、前半部に与えた代数的局所コホモロジー計算の方法をパラメータを持つ入力に対応できるように拡張することで、この問題を解決し、特異点の変形に付随したパラメータ付き代数的局所コホモロジーおよびパラメータ付きスタンダード基底を求める計算方法を確立した。これらのアルゴリズムを用いると、スタンダード基底の先頭項集合に注目したパラメータ空間の構成可能集合への分割 (stratification) と、この stratification の各 stratum に対応するパラメータ付きスタンダード基底を求めることが出来る。

2 準備

ここでは、本稿で用いる記号や基本的概念等を紹介する。 \mathbb{Q} を有理数体、 \mathbb{C} を複素数体、 \mathbb{N} を零を含む自然数全体とする。 n 変数 $x = (x_1, x_2, \dots, x_n)$ の多項式環を $K[x]$ で表す。但し、 $K = \mathbb{Q}$ (もしくは、 \mathbb{C}) とする。同様に、 n 変数 $\xi = (\xi_1, \xi_2, \dots, \xi_n)$ の多項式環を $K[\xi]$ で表す。

n 変数 x_1, \dots, x_n の単項式 $x_1^{a_1} x_2^{a_2} \cdots x_n^{a_n}$ を、 $\alpha = (a_1, \dots, a_n) \in \mathbb{N}^n$ を用いて、 x^α と表す。この時、 α を、この項 $x^\alpha = x_1^{a_1} x_2^{a_2} \cdots x_n^{a_n}$ の多重指数 または単に指数 と呼び、 $\text{mdeg}(x^\alpha)$ で表すことにする。また、 $|\alpha| = a_1 + \cdots + a_n$ をこの項の全次数と呼び、記号 $\text{tdeg}(x^\alpha)$ で表す。 ξ の単項式にたいしても同じ記号を用いる。各 $i = 1, \dots, n$ に対し、 $e_i \in \mathbb{N}^n$ を、第 i 番目の成分が 1 でそれ以外の成分は 0 となる多重指数 $e_i = (0, \dots, 0, 1, 0, \dots, 0)$ として定める。

多項式環 $K[x]$ に対し、 (\mathbb{C}^n) の原点 O に台を持つ代数的局所コホモロジー $H_{[O]}^n(K[x])$ を

$$H_{[O]}^n(K[x]) = \lim_{k \rightarrow \infty} \text{Ext}_{K[x]}^n(K[x]/\langle x_1, x_2, \dots, x_n \rangle^k, K[x])$$

で定める。ただし、 $\langle x_1, x_2, \dots, x_n \rangle$ は、 x_1, x_2, \dots, x_n が生成する極大イデアルを意味する。

いま、 X を \mathbb{C}^n の原点 O の近傍とする。この時、 $H_{[O]}^n(K[x])$ の元は、開集合対 $(X, X - \{O\})$ に対する標準的な相対被覆が定める相対 Čech コホモロジーの要素として表現することができる。逆に、多重指数

$\lambda = (l_1, l_2, \dots, l_n) \in \mathbb{N}^n$ に対し有理関数 $\frac{1}{x^{\lambda+1}} = \frac{1}{x_1^{l_1+1} x_2^{l_2+1} \dots x_n^{l_n+1}}$ が自然に定める相対 Čech コホモロジー類を $H_{[0]}^n(K[x])$ に属する代数的局所コホモロジー類と同一視して, $\left[\frac{1}{x^{\lambda+1}} \right] \in H_{[0]}^n(K[x])$ で表す。

この記号を用いると, 原点に台を持つような代数的局所コホモロジー $\psi \in H_{[0]}^n(K[x])$ はすべて

$$\psi = \sum_{\lambda} c_{\lambda} \left[\frac{1}{x^{\lambda+1}} \right] \quad (c_{\lambda} \in K, \lambda = (l_1, l_2, \dots, l_n) \in \mathbb{N}^n)$$

なる有限和により表現できる。

$H_{[0]}^n(K[x])$ は $K[x]$ 加群としての構造をもつが, x^{κ} と $\left[\frac{1}{x^{\lambda+1-\kappa}} \right]$ の積は相対 Čech コホモロジー群の定義より, 次で与えられる。

$$x^{\kappa} \left[\frac{1}{x^{\lambda+1}} \right] = \begin{cases} \left[\frac{1}{x^{\lambda+1-\kappa}} \right] & l_i \geq k_i, i = 1, \dots, n \\ 0 & \text{otherwise} \end{cases}$$

ただし, $\kappa = (k_1, \dots, k_n) \in \mathbb{N}^n, \lambda = (l_1, \dots, l_n) \in \mathbb{N}^n, \lambda + 1 - \kappa = (l_1 + 1 - k_1, \dots, l_n + 1 - k_n)$ とする ([22, 27])。

計算機上では $H_{[0]}^n(K[x])$ に属する代数的局所コホモロジー類 $\sum c_{\lambda} \left[\frac{1}{x^{\lambda+1}} \right]$ を, $\xi = (\xi_1, \xi_2, \dots, \xi_n)$ を変数とする n 変数多項式 $\sum c_{\lambda} \xi^{\lambda}$ により表現する。このような表現を代数的局所コホモロジー類の**多項式表現**とすることにする。例えば, x, y を変数とし $\psi = \left[\frac{4}{x^3 y^4} \right] + \left[\frac{5}{x^2 y^3} \right]$ を考える。このとき, ψ の多項式表現は $4\xi^2 \eta^4 + 5\xi \eta^2$ である。ただし, 変数 (ξ, η) は変数 (x, y) と対応しているとする。つまり, 2 変数では次の対応表のようになる。ただし, $c_{(l,m)} \in K$ である。

Čech 表現	多項式表現
$c_{(l,m)} \left[\frac{1}{x^{l+1} y^{m+1}} \right]$	$\longleftrightarrow c_{(l,m)} \xi^l \eta^m$
$\sum c_{(l,m)} \left[\frac{1}{x^{l+1} y^{m+1}} \right]$	$\longleftrightarrow \sum c_{(l,m)} \xi^l \eta^m$

x^{κ} と多項式表現 ξ^{λ} の乗法を記号 “*” を用いて表し, 次のように定義する:

$$x^{\kappa} * \xi^{\lambda} = \begin{cases} \xi^{\lambda-\kappa} & l_i \geq k_i, i = 1, \dots, n, \\ 0 & \text{otherwise,} \end{cases}$$

ただし, $\kappa = (k_1, \dots, k_n) \in \mathbb{N}^n, \lambda = (l_1, \dots, l_n) \in \mathbb{N}^n, \lambda - \kappa = (l_1 - k_1, \dots, l_n - k_n)$ である。

本稿では, 代数的局所コホモロジーに対し項順序を導入し, この項順序をアルゴリズムの導出に用いる。以下のように項順序を定める。

定義 1 (項順序). 二つの \mathbb{N}^n の元 $\lambda = (l_1, \dots, l_n)$ と $\lambda' = (l'_1, \dots, l'_n)$ において,

$$\left[\frac{1}{x^{\lambda'+1}} \right] \prec \left[\frac{1}{x^{\lambda+1}} \right] \quad \text{もしくは} \quad \lambda' + 1 \prec \lambda + 1$$

であるとは, $|\lambda' + 1| < |\lambda + 1|$ であるかまたは, $|\lambda' + 1| = |\lambda + 1|$ であり, $l'_i + 1 = l_i + 1, i < j, l'_j + 1 < l_j + 1$ なる $j \in \mathbb{N}$ が存在することである。

いま $\left[\frac{1}{x^{\lambda'+1}}\right] \prec \left[\frac{1}{x^{\lambda+1}}\right]$ であるとする。このことを多項式環 $K[\xi]$ においても同様に $\xi^{\lambda'} \prec \xi^{\lambda}$ と表すことにする。これは $\xi_1 \prec \xi_2 \prec \xi_3 \prec \cdots \prec \xi_n$ となる 全次数辞書式項順序 である。

代数的局所コホモロジー類 ψ

$$\psi = c_{\lambda} \left[\frac{1}{x^{\lambda+1}}\right] + \sum_{\lambda' \prec \lambda} c_{\lambda'} \left[\frac{1}{x^{\lambda'+1}}\right], \quad c_{\lambda} \neq 0,$$

に対し, $\left[\frac{1}{x^{\lambda+1}}\right]$ を主項 ($\text{ht}(\psi)$ と書く) と呼び $\left[\frac{1}{x^{\lambda'+1}}\right]$ を低階項と呼ぶ。多項式表現においても同様に, ξ^{λ} を ψ の主項 ($\text{ht}(\psi) = \xi^{\lambda}$ と書く) と呼び $\xi^{\lambda'}$ を低階項と呼ぶ。

原点 O に孤立特異点を持つ多項式 $f \in K[x]$ に対し H_f を

$$H_f = \left\{ \psi \in H_{[O]}^n(K[x]) \mid \frac{\partial f}{\partial x_1}(x)\psi = \frac{\partial f}{\partial x_2}(x)\psi = \cdots = \frac{\partial f}{\partial x_n}(x)\psi = 0 \right\}$$

で定め, 本稿では簡単のため, 特異点に付随した代数的局所コホモロジーと呼ぶことにする。これは有限次元ベクトル空間となる。その次元はミルナー数に等しい。本稿の主目的の一つはこのベクトル空間の基底を求めるアルゴリズムを紹介することである。

3 アウトライン

アルゴリズムの詳細を述べる前に全体的な流れの理解の助けとして, ここでは, 基底代数的局所コホモロジー計算 (ベクトル空間 H_f の基底計算) アルゴリズムのアウトラインを与える。

入力多項式を $f \in K[x]$ とする。 H_f の基底計算には大きく分けて 2 つのステップがある。まず **STEP 1** で, H_f の基底で単項の形をしたものを求める。次に, **STEP 2** で, 線型結合の形をしたものを求める。

OUTLINE

Input: $f \in K[x]$: 多項式,

Output: H_f の基底 (基底代数的局所コホモロジー)。

STEP 1: 単項の形をした元の計算

H_f の基底で単項の形をしたもの $\left[\frac{1}{x^{\lambda+1}}\right]$ を求める。(ただし, $\lambda \in \mathbb{N}^n$ 。)

$$\text{I.e., } \left(\frac{\partial f}{\partial x_1}\right) \left[\frac{1}{x^{\lambda+1}}\right] = \left(\frac{\partial f}{\partial x_2}\right) \left[\frac{1}{x^{\lambda+1}}\right] = \cdots = \left(\frac{\partial f}{\partial x_n}\right) \left[\frac{1}{x^{\lambda+1}}\right] = 0.$$

STEP 2: 線型結合の形をした元の計算 $\left(\sum c_{\lambda} \left[\frac{1}{x^{\lambda+1}}\right]\right)$

- (1). 主項の候補 λ を探す。
- (2). λ を主項としたとき, 低階項となる可能性となる候補 L_{λ} を構成する。
- (3). $p = \left[\frac{1}{x^{\lambda+1}}\right] + \sum_{\lambda' \in L_{\lambda}} c_{(\lambda, \lambda')} \left[\frac{1}{x^{\lambda'+1}}\right]$ ($\lambda, \lambda' \geq 0$) とする。 $c_{(\lambda, \lambda')}$ は未定係数。
- (4). 条件 $\left(\frac{\partial f}{\partial x_1}\right) p = \left(\frac{\partial f}{\partial x_2}\right) p = \cdots = \left(\frac{\partial f}{\partial x_n}\right) p = 0$ をチェックし $c_{(\lambda, \lambda')}$ を決める。すなわち, この条件からなる線型連立方程式を解く。もし, 解が存在しなければ λ は主項になることはできない。

終了条件が来るまで 1 から 4 を繰り返す。

例えば, $f = x^3 + xy^3$ で定義される E_7 特異点を考える。 $\frac{\partial f}{\partial x} = 3x^2 + y^3$, $\frac{\partial f}{\partial y} = 3xy^2$ より, **STEP 1** においてまず, 単項の形をした元を求める。計算方法の詳細は次の章でみるが, **STEP 1** から次の単項の形をした元 $\left[\frac{1}{xy}\right], \left[\frac{1}{xy^2}\right], \left[\frac{1}{x^2y}\right], \left[\frac{1}{xy^3}\right], \left[\frac{1}{x^2y^2}\right] \in H_f$ が得られる。次の **STEP 2** において線型結合の

形をした元を計算する。計算方法の詳細は次の章でみるが、**STEP 2**からは次の線型結合の形をした元 $\left[\frac{1}{xy^3}\right] - \frac{1}{3}\left[\frac{1}{x^3y}\right], \left[\frac{1}{xy^5}\right] - \frac{1}{3}\left[\frac{1}{x^3y^2}\right] \in H_f$ が得られる。これら7つの元は H_f の基底をなす代数的局所コホモロジー類である。

今後 H_f の基底計算アルゴリズムにおいて“多項式表現”を使う。計算機上で実際に代数的局所コホモロジー類を扱う際も、多項式表現を用いる。多項式表現は計算に適した表現であり、これにより計算の効率化を図れる。また、第5章でみるように、代数的局所コホモロジーからスタンダード基底を求める際も多項式表現が有効となることを予め注意しておく。

4 基底代数的局所コホモロジーの計算法

本章では、基底代数的局所コホモロジー計算アルゴリズムを紹介する。このアルゴリズムから得られるイデアルの情報を利用することでスタンダード基底が簡単に計算されることを5章でみる。同様にグレブナー基底も簡単に計算されることをみる。第3章ではアルゴリズムの概略を与えたが、この章でこのアルゴリズムの各ステップに対し、より具体的な説明を与える。この章の内容は本稿を通し最も重要な位置を占めている。

本章は3つのパートで構成される。4.1章においては単項の形となる基底の元の計算方法 (**STEP 1**) を述べ、4.2章・4.3章では線型結合の形の基底の元の計算方法 (**STEP 2**) について述べる。

本稿で紹介するアルゴリズムでは、中間データ保存のためリスト $[]$ と集合 $\{\}$ の形を使用する。リストと集合を操作するために、次の関数を使用する。

定義 2. リストは $[o_1, o_2, \dots]$ と表される。 o_1 が先頭要素である。また、何も要素を持たないリストを $[]$ と表す。ここで、 $L1, L2$ をリストとし o_1 をある要素とする。このとき、リストに作用する関数として次を定義する。

$\text{car}(L1)$: リスト $L1$ の先頭要素を返す。もし $L1 = []$ ならば、 $[]$ を返す。

$\text{cdr}(L1)$: リスト $L1$ から先頭要素を取り除いたリストを出力する。もし、 $L1 = []$ ならば、 $[]$ を返す。

$\text{cons}(o_1, L1)$: リスト $L1$ の先頭に o_1 を付け加えたリストを返す。

$\text{append}(L1, L2)$: リスト $L1$ の要素とリスト $L2$ のすべての要素を結合させたリスト $[L1 \text{ の要素の並び}, L2 \text{ の要素の並び}]$ を出力する。

集合をリストに変換する関数を list とする。集合 S に対して $\text{list}(S)$ は S の要素を成分として持つリストを意味する。この変換ではただ集合がリストになるだけで順番はどのようになってもよい。逆にリストを集合に変換する関数を set とする。 L をリストとすると、 $\text{set}(L)$ は L の成分を要素として持つ集合である。

例えば、リストとして $L = [1, 2, 3, 4, 5]$ を考える。 $\text{set}(L) = \{1, 2, 3, 4, 5\}$ であり、 $\text{car}(L) = 1$, $\text{cdr}(L) = [2, 3, 4, 5]$, $\text{cons}(8, L) = [8, 1, 2, 3, 4, 5]$, $\text{append}([7, 9], L) = [7, 9, 1, 2, 3, 4, 5]$ となる。また、集合 $S = \{0, 9, 8, 7, 6\}$ が与えられたとき、これをリストに変換するには関数 list を用いる。 $\text{list}(S) = [0, 9, 8, 7, 6]$ はリストである。

4.1 STEP 1 (単項の形の基底)

まず、3章アウトラインで述べた **STEP 1** の詳細をみる。このアルゴリズムは、 H_f の基底のうち単項の形をしたものをすべて求め、単項の形の基底を成分とする MList と、FList, GList を出力する。

(アルゴリズム) **STEP1(f)**

Input: $f: K[x]$ の多項式, (\prec : 全次数辞書式項順序,)

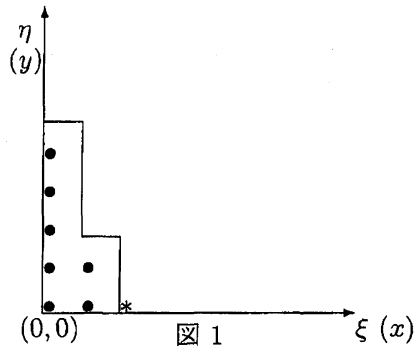
Output: MList: H_f の基底で単項の形のもの, FList: 指数のリスト, GList: リストのリスト

- (1). $\frac{\partial f}{\partial x_i} = \sum_{\alpha} a_{i,\alpha} x^{\alpha}$ ($i = 1, 2, \dots, n$) と $A = \{x^{\alpha} \in K[x] \mid \exists i, \text{ s.t. } a_{i,\alpha} \neq 0\}$ を計算する。
- (2). A から生成されるイデアル $\langle A \rangle$ の $K[x]$ における項順序 \prec に関する簡約グレブナー基底 G を計算する。
- (3). 項順序に関して G 内で最小な元 t を選択する。 $\text{FList} \leftarrow \text{list}(\{\text{mdeg}(t)\})$
- (4). 順序付けられたリストのリスト $\text{GList} = [G_{i_1}, G_{i_2}, \dots, G_{i_s}]$ を作成する。ただし、各 $1 \leq j \leq s$ と $1 \leq k \leq l_j$ において、 $G_{i_j} = [\text{mdeg}(r_{j,1}), \text{mdeg}(r_{j,2}), \dots, \text{mdeg}(r_{j,l_j})]$, $r_{j,k} \in G \setminus \{t\}$, $\text{tdeg}(r_{j,k}) = i_j$, $i_1 < i_2 < \dots < i_s$, $r_{j,1} \prec r_{j,2} \prec \dots \prec r_{j,l_j}$ とする。
- (5). $\text{MList} \leftarrow K[x]/\langle G \rangle$ の標準単項 (standard monomial) を計算する。

MList に属す単項式 p は構成の仕方から明らかに、条件 $\left(\frac{\partial f}{\partial x_1}\right) * p = \left(\frac{\partial f}{\partial x_2}\right) * p = \dots = \left(\frac{\partial f}{\partial x_n}\right) * p = 0$ を満たす (従って、 MList は H_f の基底の部分集合である)。逆に、この条件を満たす単項式はすべて MList に属す単項式 (の定数倍) であることを容易に確かめることができる。アルゴリズム内で FList , GList を構成し出力させるようにしてあるが、これらは引き続き計算において有効に用いられる。ここではアルゴリズム **STEP1** が具体的にどのように働くか次の例で示す。

例 3. 2 変数多項式 $f = x^3 + y^6 + x^2y^2 \in K[x, y]$ を考える。項順序は定義 1 にあるように全次数辞書式項順序 \prec であり、 $y \prec x$ とする。

- (1). まず、 $\frac{\partial f}{\partial x} = 3x^2 + 2xy^2$, $\frac{\partial f}{\partial y} = 6x^5 + 2x^2y$ と $A = \{x^2, xy^2, x^5, x^2y\}$ を得る。
- (2). $\langle A \rangle$ の簡約グレブナー基底として $G = \{x^2, xy^2, y^5\}$ を得る。
- (3). 項順序 \prec に関して、 x^2 は G で最小の元であるので、
 $\text{FList} = [\text{mdeg}(x^2)] = [(2, 0)]$ となる。
- (4). 全次数 3 の元は xy^2 であり、全次数 5 の元は y^5 である。したがって、 $\text{GList} = [[(1, 2)], [(0, 5)]]$ となる。
- (5). 最後に、 $K[x, y]/\langle G \rangle$ の標準単項 MList を求める。このとき、 $\text{MList} = [1, \xi, \eta, \eta^2, \xi\eta, \eta^3, \eta^4]$ となる。図 1 において、記号 “•” は MList の成分の指数を表す。記号 “*” は FList の成分の指数を表す。



4.2 主項の決め方

4.2 章と 4.3 章では、3 章アウトラインの **STEP 2** について考える。**STEP 2** では、線型結合の形 $\sum c_{\lambda} \left[\frac{1}{x^{\lambda+1}} \right]$ (多項式表示は $\sum c_{\lambda} \xi^{\lambda}$) を計算しなければならない。この計算のアルゴリズムは大きく分けて 2 つのパートに分かれる。1 つ目は、主項を決定するパートであり、4.2 章で述べる。2 つ目は、低階項候補の求め方、決め方のパートであり 4.3 章で述べる。計算アルゴリズムの構成のため、まずいくつかの先行結果を紹介する。今、 Λ_H を H_f における主項の指数の集合とし、 $\Lambda_H^{(\lambda)}$ を $\lambda' \prec \lambda$ となる Λ_H の部分集合とする。I.e.,

$$\Lambda_H = \{\lambda \in \mathbb{N}^n \mid \exists \psi \text{ s.t. } \text{ht}(\psi) = \xi^{\lambda}, \psi \in H_f\}, \quad \Lambda_H^{(\lambda)} = \{\lambda' \in \Lambda_H \mid \lambda' \prec \lambda\}.$$

次の結果は本稿のアルゴリズム内でよく使う重要なものである。

補題 4 ([25, 29]). ψ が H_f に含まれているならば, 各 $j = 1, 2, \dots, n$ に対して $x_j \psi$ も H_f に含まれる。

補題 5 ([25, 29]). $\lambda = (l_1, \dots, l_n)$ を \mathbb{N}^n の元とする。もし, $\lambda \in \Lambda_H$ ならば, そのとき, 各 $j = 1, 2, \dots, n$ に対して, $\lambda - e_j = (l_1, l_2, \dots, l_{j-1}, l_j - 1, l_{j+1}, \dots, l_n)$ は $\Lambda_H^{(\lambda)}$ に含まれる。

補題 6 ([25, 29]). $\lambda = (l_1, \dots, l_n)$ を \mathbb{N}^n の元とする。もし, $\lambda \notin \Lambda_H$ ならば, そのとき, $(\lambda + \mathbb{N}^n) \cap \Lambda_H = \emptyset$ である。(I.e., アルゴリズム内の記号を使うと $\exists \alpha \in \text{FList}, x^\alpha * \xi^\lambda \neq 0$ である。)

さて, いまここで補題 6 に注目する。**STEP1** の出力として与えられる FList の成分 (一個のみ) は MList に属さない単項式のうち最も項順序が低い単項式の指数である。従って, この指数は決して Λ_H に属さないことが分かる。ここで補題 6 を使うと, $\Lambda_H \setminus \text{set}(\text{MList})$ に属す指数のうち項順序が最も低いものは GList に属すリストに含まれ, さらに FList に属す指数は, その主項に対する低階項候補となることが容易にわかる。主項候補とそれに対する低階項候補に関し, これに類した議論を計算の各過程で行うことで計算アルゴリズムを導出した。これは動的なアルゴリズムであり, 計算のために複数のリストを計算の進行と共に更新を繰り返しながら同時に扱っていくことになる。このアルゴリズムで用いる主なリストを以下に与える。

MList: 成分は $K[\xi]$ の要素で, H_f の基底の中で単項の形をしたもの。(STEP 1 を見よ)

SList: 成分は $K[\xi]$ の要素で, H_f の基底の中で線型結合の形をしたもの。

TList: 成分は \mathbb{N}^n の要素で, SList の要素の主項の指数。

FList: 成分は \mathbb{N}^n の要素で, 主項の候補になったが主項にならなかったもの。

LList: 成分は \mathbb{N}^n の要素 SList の要素すべての低階項の指数。

GList: リストのリスト。STEP 1 (4) で定義されるリスト。

CT: 成分は \mathbb{N}^n の要素で主項候補の指数。すべての要素の全次数は等しい。

TT: CT に含まれる項のうち実際に主項となる項の指数。すべての要素の全次数は等しい。

CL: 成分は \mathbb{N}^n の要素で (与えられた主項に対する) 低階項候補の指数。

従って, 上の記号を使うと H_f の基底は最終的には $\text{append}(\text{MList}, \text{SList})$ で与えられる。低階項候補のリスト CL を決めるとき, 上記以外のリスト EL, LL, UU, RR を補助的に用いる。アルゴリズムの整合性により, これらの記号, EL, LL, UU, RR を次のアルゴリズム Head_main 内においても使用するが, 低階項候補リストに関する説明は 4.3 章で行うのでここでは説明を省く。

次のアルゴリズム Head_main は, STEP 2 での基底代数的局所コホモロジー計算の中核をなすアルゴリズムである。このアルゴリズムは全次数辞書式項順序に沿って, 主項の全次数の低いものから決定していく。そのため, リスト CT, TT のすべての要素の全次数は等しくなる。計算の効率化を考えれば全次数の低いものからボトムアップで決定していくことは自然な流れである。

(アルゴリズム) Head_main(GList, FList, MList)

Input: GList, FList, MList: STEP 1 から得られらリスト

Output: SList, LList, TList, FList

(0). $\text{TT} \leftarrow []$; $\text{CT} \leftarrow \text{car}(\text{GList})$; $\text{GList} \leftarrow \text{cdr}(\text{GList})$; $\text{TList} \leftarrow []$; $\text{ML} \leftarrow \{\text{mdeg}(g) | g \in \text{MList}\}$

$\text{LList} \leftarrow \text{FList}$; $\text{EL} \leftarrow \text{FList}$; $\text{LL} \leftarrow []$; $\text{RR} \leftarrow []$; $\text{UU} \leftarrow []$ (◇1)

(1). if $\text{CT} \neq []$ then

$\gamma \leftarrow \text{car}(\text{CT})$, $\text{CT} \leftarrow \text{cdr}(\text{CT})$ /* CT から最小元 γ を選ぶ */

else if $\text{CT} = []$ then

$(\text{CT}, \text{GList}) \leftarrow \text{Head_candidate}(\text{TT}, \text{CT}, \text{GList})$ (*1) /*主項候補を作る*/


```

if CT = [] then (*2) /* 終了条件。(候補が無いとき) */
  return(SList, TList, LList, FList)

```

```

end-if

```

```

TT ← []; γ ← car(CT), CT ← cdr(CT) /* CT から最小元 γ を選ぶ */

```

```

end-if

```

(2). ここで ξ^γ を主項候補とする。/*低階項候補リスト CL を作成する*/

```

[CL, UU, EL] ← low-candidate(γ, LList, LL, UU, RR, EL, ML, TList) (◇2)

```

(3). $p = \xi^\gamma + \sum_{\lambda \in CL} c_\lambda \xi^\lambda$ と置く。(ここで $c_\lambda \in K$ 。)

条件 $\left(\frac{\partial f}{\partial x_1}\right) * p = \left(\frac{\partial f}{\partial x_2}\right) * p = \dots = \left(\frac{\partial f}{\partial x_n}\right) * p = 0$ から c_λ の連立方程式を作り解く。

if 解 c_λ が存在する then $Z \leftarrow 1$

```

  TT ← cons(γ, TT); p' ← p の  $c_\lambda$  に解を代入する

```

```

  TList ← cons(γ, TList); SList ← cons(p', SList) /* p' は基底の元である */

```

```

else  $Z \leftarrow 0$ ; p' ← 0

```

```

  FList ← cons(γ, FList) /*  $\xi^\gamma$  は基底の主項ではない */

```

```

end-if

```

```

[LList, LL, RR, EL] ← renewlow(Z, γ, p', LList, EL) (◇3)

```

終了条件 (*2) が来るまで (1) から (3) を繰り返せ

Remark: 前述したように、低階項に関する行 (◇1), (◇2), (◇3) は 4.3 章で説明する。I.e., プロシジャ low-candidate, renewlow は 4.3 章にある。ここでは、これらの出力のうち CL, LList の動作を見ればアルゴリズムの流れは理解できる。(CL, LList の意味は前述している。)

アルゴリズムの (1) において、もし $CT = []$ ならば、(*1) において CT を更新しなければならない。主項候補の指数からなるリスト CT の更新を行う際、補題 4, 5, 6 を用いるが、そのために次の 2 つの関数 “nb” と “cf” を使用する。

(関数) nb(TT)

Input: TT,

Output: S: リスト

```

S ← []

```

```

while TT ≠ [] then

```

```

  τ ← car(TT); TT ← cdr(TT)

```

```

  for i from 1 to n do

```

```

    α ← τ + ei

```

```

    if α ∉ S then

```

```

      S ← cons(α, S)

```

```

    end-if

```

```

  end-for

```

```

end-while

```

```

return(S)

```

(関数) cf(L, FList)

Input L: \mathbb{N}^n のリスト, FList,

Output S: リスト

```

S ← []

```

```

while L ≠ [] do t ← 1

```

```

  α ← car(L); L ← cdr(L); W ← FList

```

```

  while W ≠ [] do

```

```

    κ ← car(W); W ← cdr(W)

```

```

    if  $x^\kappa * \xi^\alpha \neq 0$  then t ← 0; break

```

```

  end-if

```

```

  end-while

```

```

  if t = 1 then S ← cons(α, S) end-if

```

```

end-while

```

```

return(S)

```

例えば, $L = [(1, 2), (3, 1), (3, 4), (1, 8)]$, $FList = [(2, 3), (4, 0)]$ とする。このとき、上の関数を実行させると、

$\text{nb}(L) = [(2, 2), (1, 3), (4, 1), (3, 2), (4, 4), (3, 5), (2, 8), (1, 9)]$, $\text{cf}(L, \text{FList}) = [(1, 2), (3, 1), (1, 8)]$ となる。

さて, $\text{CT} = []$ であり, これから CT の更新を行うとしよう。この場合, 状況として次の 4 つの **Case** がある。

Case (i) $\text{TT} = []$, $\text{GList} = []$. **Case (ii)** $\text{TT} = []$, $\text{GList} \neq []$.

Case (iii) $\text{TT} \neq []$, $\text{GList} = []$. **Case (iv)** $\text{TT} \neq []$, $\text{GList} \neq []$.

次のプロシジャー Head_candidate は各 **Case** においてどのように CT を更新するかを決定する。例えば, もし **Case (i)** “ $\text{TT} = []$ and $\text{GList} = []$ ” の状況ならば, 上のアルゴリズム Head_main は停止する。

(プロシジャー) $\text{Head_candidate}(\text{TT}, \text{CT}, \text{GList})$

Input: $\text{TT}, \text{CT}, \text{GList}$: Head_main からのリスト

Output: CT, GList : リスト (新しい主項候補)

Case (i) $\text{TT} = []$, $\text{GList} = []$: $\text{return}([], [])$ /* 終了 */

Case (ii) $\text{TT} = []$, $\text{GList} \neq []$: $\text{CT} \leftarrow \text{car}(\text{GList})$; $\text{GList} = \text{cdr}(\text{GList})$; $\text{return}(\text{CT}, \text{GList})$

Case (iii) $\text{TT} \neq []$, $\text{GList} = []$: $\text{NT} \leftarrow \text{nb}(\text{TT})$; $\text{CT} \leftarrow \text{cf}(\text{NT}, \text{FList})$; $\text{return}(\text{CT}, \text{GList})$

Case (iv) $\text{TT} \neq []$, $\text{GList} \neq []$:

リスト TT の成分の全次数を $dt \in \mathbb{N}$ とする。(TT のすべての要素の全次数は等しい。)

リスト $\text{car}(\text{GList})$ の全次数を $dg \in \mathbb{N}$ とする。(car(GList) のすべての要素の全次数は等しい。)

一般的に, $dg > dt$ ($dg \neq dt$) である。

if $dg - dt > 1$ **then** $\text{NT} \leftarrow \text{nb}(\text{TT})$; $\text{CT} \leftarrow \text{cf}(\text{NT}, \text{FList})$; $\text{return}(\text{CT}, \text{GList})$

end-if

if $dg - dt = 1$ **then** $\text{CT} \leftarrow \text{cf}(\text{nb}(\text{TT}), \text{FList}) \cup \text{car}(\text{GList})$; $\text{GList} \leftarrow \text{cdr}(\text{GList})$

$\text{return}(\text{CT}, \text{GList})$

end-if

(注) リストが生成された後, すべてのリストは項順序 \prec に関して順序付けられたリストとして並び変えられているものとする。

全次数の低いものから決定し終了条件が満たされるまで, 全次数を上げながら計算を続ける。アルゴリズムがどのように働くか例を通して見てみよう。

例 7. 例 3 と同じ問題を考える。2 変数多項式 $f = x^3 + y^6 + x^2y^2 \in K[x, y]$ を見る。例 3 において, $\text{FList} = [(2, 0)]$, $\text{GList} = [(1, 2), (0, 5)]$ を得ている。ここで, H_f の基底を計算するためアルゴリズム Head_main を適用する。低階項候補作成・決定 (($\diamond 1$), ($\diamond 2$), ($\diamond 3$)) については後ほど, 例 9 において述べることにし, ここではアルゴリズム内にある ($\diamond 1$), ($\diamond 2$), ($\diamond 3$) の働きについて触れないことにする。

1-0. $\text{TT} = []$; $\text{CT} = \text{car}(\text{GList}) = [(1, 2)]$; $\text{GList} = \text{cdr}(\text{GList}) = [(0, 5)]$ 。

1-1. アルゴリズム Head_main (1) より, $\gamma = \text{car}(\text{CT}) = (1, 2)$, $\text{CT} = \text{cdr}(\text{CT}) = []$ である。今, 主項の候補は $\xi\eta^2$ である。(2) において, プロシジャー low_candidate は $\xi\eta^2$ の低階項候補として CL を出力する。(4.3 章において CL の計算は議論する。)(3) において, $p_1 = \xi\eta + \sum_{\lambda \in \text{CL}} c_\lambda \xi^\lambda$ とおき, 条件 $\left(\frac{\partial f}{\partial x}\right) * p_1 = \left(\frac{\partial f}{\partial y}\right) * p_1 = 0$ から生成される連立方程式 E_q を解くことで未定係数を求め, $p_1 = \xi\eta^2 - \frac{2}{3}\xi^2$ を得る。ここで各リストを更新すると, $\text{TT} = \text{cons}((2, 1), \text{TT}) = [(2, 1)]$, $\text{SList} = \text{cons}(p_1, \text{SList}) = [p_1]$ となる。図 8-1 において, 記号 “ \bullet ” は MList の成分の指数を表し, “ $*$ ” は FList の成分を表す。また, p_1 は $\text{ht}(p_1)$ を表している。

2-0. 現在 $CT = []$, $TT = [(1, 2)]$, $GList = [(0, 5)]$ である。この場合、プロシジャー **Head.candidate Case (iv)** を適用する。 TT の成分の全次数 3, $\text{car}(GList)$ の成分の全次数 5 より, $NT = \text{nb}(TT) = [(1, 3), (2, 2)]$, $CT = \text{cf}(NT, FList) = [(1, 3)]$ となる。

2-1. アルゴリズム **Head.main** (1) よりリストを更新すると, $TT = []$, $\gamma = \text{car}(CT) = (1, 3)$, $CT = []$ となる。今, 主項候補は $\xi\eta^3$ である。プロシジャー **low-candidate** は $\xi\eta^3$ における低階項候補 CL を出力する。 $p = \xi\eta^3 + \sum_{\lambda \in CL} c_\lambda \xi^\lambda$ と置き条件 $\left(\frac{\partial f}{\partial x}\right) * p = \left(\frac{\partial f}{\partial y}\right) * p = 0$ から生成される連立方程式 Eq を解く。 Eq は解を持たないので, $\xi\eta^3$ は主項になり得ない。リストを更新する。 $FList = \text{cons}((1, 3), FList) = [(1, 3), (2, 0)]$ 。

3-1. $CT = \text{car}(GList) = [(0, 5)]$ より, 次の主項の候補は $(0, 5)$ である。リストを更新すると, $CT = \text{cdr}(CT) = []$, $GList = \text{cdr}(GList) = []$ となる。プロシジャー **low-candidate** は η^5 における低階項候補 CL を出力する。 $p_2 = \eta^5 + \sum_{\lambda \in CL} c_\lambda \xi^\lambda$ と置き条件 $\left(\frac{\partial f}{\partial x}\right) * p = \left(\frac{\partial f}{\partial y}\right) * p = 0$ から生成される連立方程式 Eq を解き, $p_2 = \eta^5 + \frac{9}{2}\xi\eta^3 - 3\xi^2\eta$ を得る。リストを更新すると, $TT = \text{cons}((0, 5), TT) = [(0, 5)]$, $SList = \text{cons}(p_2, SList) = [p_2, p_1]$ となる。今の状況を図に表すと図 8-2 となる。

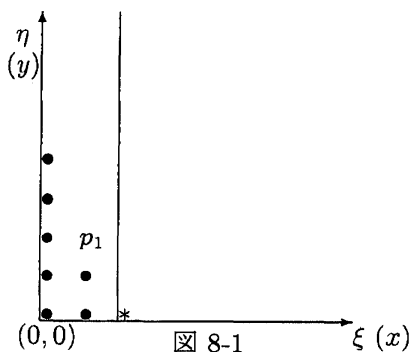


図 8-1

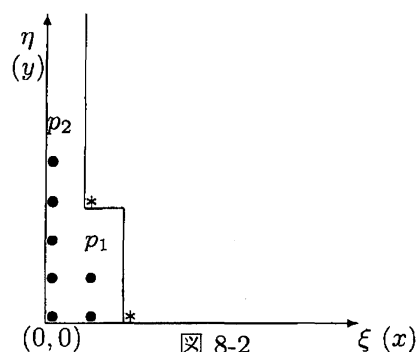


図 8-2

4-0. $CT = []$, $TT = [(0, 5)]$, $GList = []$ である。この場合、プロシジャー **Head.candidate Case (iii)** を適用する。 $NT = \text{nb}((TT)) = [(0, 6), (1, 5)]$ より $CT = \text{cf}(NT, FList) = [(0, 6)]$ となる。

4-1. アルゴリズムの **Head.main** (1) よりリストを更新すると, $TT = []$, $\gamma = \text{car}(CT) = (0, 6)$, $CT = \text{cdr}(CT) = []$ となる。主項候補は η^6 である。プロシジャー **low-candidate** は η^6 における低階項候補 CL を出力する。 $p_3 = \eta^6 + \sum_{\lambda \in CL} c_\lambda \xi^\lambda$ と置き条件 $\left(\frac{\partial f}{\partial x}\right) * p = \left(\frac{\partial f}{\partial y}\right) * p = 0$ から生成される連立方程式 Eq を解き, $p_3 = \eta^6 + \frac{9}{2}\xi\eta^4 - 3\xi^2\eta^2 + 2\xi^3$ を得る。リストを更新すると, $TT = \text{cons}((0, 6), TT) = [(0, 6)]$, $SList = \text{cons}(p_3, SList) = [p_3, p_2, p_1]$ となる。今の状況を図に表すと図 8-3 となる。

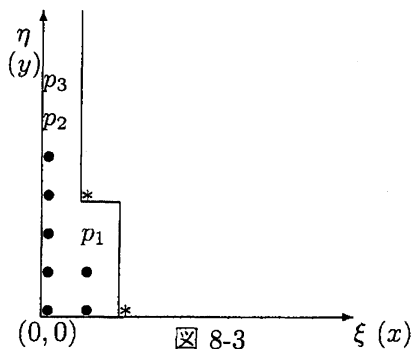


図 8-3

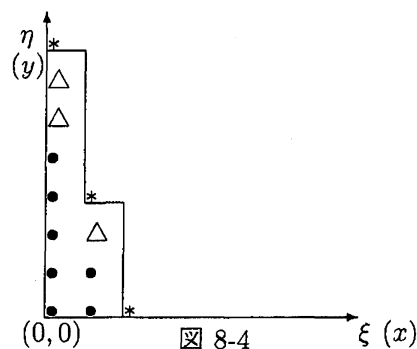


図 8-4

5-0. $CT = []$, $TT = [(0, 6)]$, $GList = []$ である。この場合、プロシジャー **Head.candidate Case (iii)** を適用する。 $NT = \text{nb}((TT)) = [(0, 7), (1, 6)]$ より $CT = \text{cf}(NT, FList) = [(0, 7)]$ となる。

5-1. アルゴリズム Head.main (1) よりリストを更新すると, $TT = []$, $\gamma = \text{car}(CT) = (0, 7)$, $CT = \text{cdr}(CT) = []$ となる。主項候補は η^7 である。プロシジャー low-candidate は η^7 における低階項候補 CL を出力する。 $p = \eta^7 + \sum_{\lambda \in \text{CL}} c_\lambda \xi^\lambda$ と置き条件 $\left(\frac{\partial f}{\partial x}\right) * p = \left(\frac{\partial f}{\partial y}\right) * p = 0$ から生成される連立方程式 E_q を解くと, E_q は解を持たない。したがって, η^7 は主項になり得ない。リストを更新する。
 $\text{FList} = \text{cons}((0, 7), \text{FList}) = [(0, 7), (1, 3), (2, 0)]$ 。

6-0. ここで, $TT = []$, $\text{GList} = []$ であるので計算を終了する。したがって, f に付随する基底代数的局所コホモロジー (H_f の基底) は $\text{append}(\text{MList}, \text{SList})$ であり, 10 個の元からなる。最終的な結果を図示すると図 8-4 になる。ただし, 図 8-4 では SList の成分の主項を記号 Δ を用いて表した。

4.3 低階項の決め方

ここでは低階項の決定法について述べる。ここでの目的は 4.2 章で見たアルゴリズム Head.main 内で使用したプロシジャー renewlow, low-candidate を構成することである。先行研究として H_f の基底を求めるアルゴリズムが存在する [1, 25, 26]。本稿と先行研究とのおおきな違いは低階項候補の構成の仕方にある。ここで紹介するプロシジャーは, 計算過程で現れる中間生成物 MList , FList , TList , LList , SList の情報を有効に利用し低階項候補を絞り込むことで先行研究の結果を大きく改良し効率化したものである。プロシジャーを述べる前に低階項の性質について次の重要な補題を紹介する。ここでは, $\text{ML} = \{\text{mdeg}(g) | g \in \text{MList}\}$ とする。

補題 8. ψ を H_f の元とし, $\left[\frac{1}{x_1^{l_1+1} \cdots x_n^{l_n+1}} \right]$ を ψ の低階項とする。 H_f の基底を計算する間 (アルゴリズム Head.main において), 項

$$\left[\frac{1}{x_1^{l_1} x_2^{l_2+1} \cdots x_n^{l_n+1}} \right], \left[\frac{1}{x_1^{l_1+1} x_2^{l_2} x_3^{l_3+1} \cdots x_n^{l_n+1}} \right], \dots, \left[\frac{1}{x_1^{l_1+1} x_2^{l_2+1} \cdots x_{n-1}^{l_{n-1}+1} x_n^{l_n}} \right]$$

の指数は次のいずれかに所属する;

- (1) LList , (2) TList , (3) ML , (4) $[0]$ (零になる)

多項式表現を使えば, この補題は次のように書くことができる。 $\xi_1^{l_1} \cdots \xi_n^{l_n}$ を ψ の低階項とする。このとき, 指数

$$(l_1 - 1, l_2, \dots, l_n), (l_1, l_2 - 1, \dots, l_n), \dots, (l_1, \dots, l_{n-1} - 1, l_n), (l_1, \dots, l_{n-1}, l_n - 1)$$

は, (i) $\text{set}(\text{ML}) \cup \text{set}(\text{TList}) \cup \text{set}(\text{LList})$ に所属する。もしくは, (ii) ある成分が零になる。

次の関数 “cd” は補題 8 を利用する際に用いる。

(関数) $\text{cd}(\text{NL}, \text{ML}, \text{TList}, \text{LList})$

Input: NL : \mathbb{N}^n の成分を持つリスト, ML , TList , LList

Output: D : リスト

$D \leftarrow []$

while $\text{NL} \neq []$ **do**

$\lambda \leftarrow \text{car}(\text{NL}); \text{NL} \leftarrow \text{cdr}(\text{NL})$

if $(\lambda \in \text{set}(\text{ML}) \cup \text{set}(\text{TList}) \cup \text{set}(\text{LList}))$ **or** $(\exists i \in \mathbb{N} \text{ s.t. } \lambda \text{ の第 } i \text{ 成分が } 0)$ **then**

$D \leftarrow \text{cons}(\lambda, D)$ **end-if**

end-while

return (D)

ここで低階項の元について考える。もし、低階項の指数が TList, ML に含まれていると、 $K[[x]]/J$ を考えているので、その元は SList, MList によって簡約化することができる。すなわち、もともと SList の持つすでにある低階項 LList の線型結合で表すことができる。したがって、ボトムアップで計算した結果、得られている LList は低階項候補になることが分かる。以上より、低階項候補 CL は「新しく候補になる項」と LList を合わせたものとなる。

低階項候補生成アルゴリズムにおいて、次のリストを用いて中間データを一時的に保存し利用する。(計算は項順序の小さいものからする。)

- EL: 新しく候補になる項のリストで、LList には属さないもの。
- LL: EL に属する項で現在求めたコホモロジーの低階項として真に現れた項すべて。
- RR := list(set(EL) \ set(LL))。
- UU: nb(LL) の要素のうち、項順序が主項候補より大きくなるため、EL に加えなかった項すべて (プロシジャー low-candidate を見よ)。

(注) CL = append(EL, LList) となる。

アルゴリズム Head_main (3) (◇ 3) において、リスト LList, UU, RR, EL を更新しなければならないが、連立方程式に解が存在する場合 ($Z = 1$) と解が存在しない場合 ($Z = 0$) とで更新の仕方が異なる。

(プロシジャー) renewlow(LList, EL, Z, γ , p')

Input: LList, EL, Z: 0 もしくは 1, γ : N^n の元, $p': K[\xi]$ の多項式 (コホモロジー)
(すべての記号は Head_main (◇3) と同じとする。)

Output: LList, LL, RR, EL,

if $Z = 0$ then

LL \leftarrow []; EL \leftarrow cons(γ , EL); RR \leftarrow []

else

LL \leftarrow list({ p' のすべての低階項 } \cap set(EL)) (♣1)

if LL \neq [] then

LList \leftarrow append(LList, LL); RR \leftarrow list(set(EL) \ set(LL)); EL \leftarrow []

end-if

end-if

return(LList, LL, RR, EL)

プロシジャー renewlow の (♣1) において、LL = list({ p' のすべての低階項 } \cap set(EL)) であるので、LL は LList に存在しない項である。したがって、LL \neq [] なら、補題 8 より、次の主項候補のために新しい低階項候補を生成しなければならない。そのため、次のプロシジャーは LL = [] か LL \neq [] かにより動作が異なる。ここでは関数 "nb" と "cd" が用いられている。

(プロシジャー) low-candidate(γ , LList, LL, UU, RR, EL, ML, TList)

Input: $\gamma \in N^n$, LList, LL, UU, RR, EL, ML, TList : アルゴリズム Head_main からのリスト,

Output: CL, UU, EL: リスト, (CL は主項 γ の低階項候補)

(Case 1) if LL = [] then

if UU \neq [] then

E \leftarrow { $\alpha | \gamma \succ \alpha$ (α は γ より低い), $\alpha \in UU$ }

UU \leftarrow list(set(UU) \ cd(E)); EL \leftarrow append(list(cd(E)), EL)

end-if

```

    end-if
    CL ← append(EL, LList)
    return(CL, UU, EL)
  (Case 2) if LL ≠ [] then
    /* RR と LL からリスト EL, UU を生成する。 */
    E ← {α | γ ≻ α, (α は γ より低い), α ∈ UU}
    UU ← list((set(UU) \ E) \ {γ}) /*if γ ∈ UU */
    RR ← append(list(E), RR)
    NL ← set(nb(LL))
    B ← {β | β ≻ γ (γ は n β より低い), β ∈ NL}
    UU ← append(list(B), UU)
    D ← cd(list(NL \ B), ML, TList, LList)
    EL ← append(list(D \ (D ∩ set(RR))), RR)
  end-if
  CL ← append(EL, LList)
  return(CL, UU, EL)

```

プロシジャー renewlow と low-candidate がどのように働くか例を与える。

例 9. 例 3, 例 7 と同じ問題を考え, $f = x^3 + y^6 + x^2y^2 \in K[x, y]$ とおく。例 7 において, 主項の決め方についてみた。ここでは, 低階項の決め方についてみる。

0. アルゴリズム Head_main(◇1) においてリストを初期化する。LList = [], EL = [(2, 0)] (= FList), RR = [], UU = [], LL = []。
1. (1, 2) は主項候補である。今, LL = [], UU = [] なので, プロシジャー low-candidate (Case 1) より, CL = append(EL, LList) = [(2, 0)] である。 $p = \xi\eta^2 + c_{(2,0)}\xi^2$ と置き, 条件 $\left(\frac{\partial f}{\partial x}\right) * p = \left(\frac{\partial f}{\partial y}\right) * p = 0$ をチェックする。すなわち, 線型の連立方程式 Eq: $\left(\frac{\partial f}{\partial x}\right) * p = 3c_{(2,0)} + 2 = 0, \left(\frac{\partial f}{\partial y}\right) * p = 0 = 0$ を解き, $c_{(2,0)} = -\frac{2}{3}$ 。すなわち, $p = \xi\eta^2 - \frac{2}{3}\xi^2$ を得る。アルゴリズム Head_main (3) において $Z = 1$ となり, プロシジャー renewlow によりリストを更新する。LL = [(2, 0)], EL = [], LList = [(2, 0)]。
2. (1, 3) は主項候補である。今, LL = [(2, 0)] なので, プロシジャー low-candidate (Case 2) より, NL = nb(LL) = [(3, 0), (2, 1)] であり, これは他のリストにより簡約されないで, そのまま EL = NL となり CL = append(EL, LList) = [(3, 0), (2, 1), (2, 0)] となる。 $p = \xi\eta^3 + c_{(3,0)}\xi^3 + c_{(2,1)}\xi^2\eta + c_{(2,0)}\xi^2$ と置き, 条件をチェックすると連立方程式 Eq: $c_{(3,0)} = 0, c_{(2,1)} + \frac{2}{3} = 0, 3c_{(2,0)} = 0, -\frac{4}{3} = 0$ を得る。Eq は解を持たない。アルゴリズム Head_main (3) において $Z = 0$ を得る。FList = cons((1, 3) FList) = [(1, 3), (2, 0)]。プロシジャー renewlow により, リストを更新する。LL = [], EL = cons((1, 3), EL) = [(1, 3), (3, 0), (2, 1)]。
3. (0, 5) は主項候補である。低階項候補は CL = append(EL, LList) = [(1, 3), (3, 0), (2, 1), (2, 0)] である。 $p = \eta^5 + c_{(1,3)}\xi\eta^3 + c_{(3,0)}\xi^3 + c_{(2,1)}\xi^2\eta + c_{(2,0)}\xi^2$ と置き, 条件をチェックすると連立方程式 Eq: $c_{(3,0)} = 0, c_{(2,1)} + \frac{2}{3}c_{(1,3)} = 0, 3c_{(2,0)} = 0, 6 - \frac{4}{3}c_{(1,3)} = 0$ を得る。このとき, Eq は解を持ち, $p = \eta^5 + \frac{9}{2}\xi\eta^3 - 3\xi^2\eta$ を得る。リストを更新する。LL = [(1, 3), (2, 1)], LList = [(1, 3), (2, 1), (2, 0)], RR = [(3, 0)]。

4. $(0, 6)$ は主項候補である。NL = nb(LL) = $[(2, 3), (1, 4), (3, 2), (2, 2)]$ であり, $\text{cd}(\text{NL}, \text{ML}, \text{TList}, \text{LList}) = [(1, 4), (2, 2)]$ より, 低階項候補は CL = $[(1, 4), (2, 2), (1, 3), (3, 0), (2, 1), (2, 0)]$ である。 $p = \eta^6 + c_{(1,4)}\xi\eta^4 + c_{(2,2)}\xi^2\eta^2 + c_{(1,3)}\xi\eta^3 + c_{(3,0)}\xi^3 + c_{(2,0)}\xi^2$ と置き, 条件をチェックし連立方程式 Eq: $c_{(3,0)} + \frac{2}{3}c_{(2,2)} = 0, c_{(1,4)} - \frac{9}{2} = 0, c_{(2,2)} + 3 = 0, c_{(2,1)} + \frac{2}{3}c_{(1,3)} = 0$ を得る。Eq は解を持ち, $p = \eta^6 + \frac{9}{2}\xi\eta^4 - 3\xi^2\eta^2 + 2\xi^3$ を得る。リストを更新する。LL = $[(1, 4), (2, 4)]$, LList = $[(1, 4), (2, 2), (1, 3), (2, 1), (2, 0)]$, RR = $[(3, 0)]$ 。
5. $(0, 7)$ は主項候補である。今までと同様に計算すると低階項候補として CL = $[(1, 5), (2, 3), (1, 4), (4, 0), (3, 1), (2, 2), (1, 3), (3, 0), (2, 1), (2, 0)]$ を得る。このとき, $p = \sum_{(i,j) \in \text{CL}} c_{(i,j)}\xi^i\eta^j$ と置き, 条件をチェックしても, 解は存在しない。FList = cons($(0, 7)$, FList) = $[(7, 0), (1, 3), (2, 0)]$

準備が全て整ったので, H_f の基底を求めるアルゴリズムを構成できる。次の 5 章でスタンダード基底計算, グレブナー基底計算, イデアルメンバーシップ問題を解くアルゴリズムについて論じるが, これらのアルゴリズムでは, TList, LList, FList が持つ情報等も利用する。従って, 次のアルゴリズム ALC では, MList, SList 以外に TList, LList, FList も出力させるようにした。このアルゴリズム ALC は既に, 計算機代数システム Risa/Asir ([23]) に実装済みである。

(アルゴリズム) ALC(f)

Input: $f: K[x]$ の多項式,

Output: MList, SList, TList, LList, FList. (H_f の基底は append(MList, SList))

(MList, FList, GList) \leftarrow STEP1(f)

(SList, TList, LList, FList) \leftarrow Head.main(GList, FList, MList)

return(MList, SList, TList, LList, FList)

H_f の基底として MList と SList のみを出力するようなプログラム cohomology も作成した。その出力例を以下に記す。

例 10. 多項式 $f = x^3 + y^7 + xy^5$ によって定義される E_{12} 特異点を考える。最後の引数として 0 をプログラムに与えた場合, このプログラムは, H_f の基底を多項式表現により出力する。

```
[467] cohomology(x^3+y^7+x*y^5,[x,y],1,0,0);
[[1,y,x,y^2,y*x,y^3,y^2*x,y^4,y^3*x],[-1/3*x^2+y^5,-1/3*y*x^2-7/5*y^4*x+y^6,7/15*x^3-1/3*y^2*x^2-7/5*y^5*x+y^7]]
```

出力の第 1 リストは MList を意味し, 第 2 リストは SList を意味する。また, 最後の引数として 1 を与えた場合, プログラム cohomology は H_f の基底を Čech 表現により出力する。

```
[468] cohomology(x^3+y^7+x*y^5,[x,y],1,0,1);
[[y^(-1)*x^(-1),y^(-2)*x^(-1),y^(-1)*x^(-2),y^(-3)*x^(-1),y^(-2)*x^(-2),y^(-4)*x^(-1),y^(-3)*x^(-2),y^(-5)*x^(-1),y^(-4)*x^(-2)], [y^(-6)*x^(-1)-1/3*y^(-1)*x^(-3),y^(-7)*x^(-1)-7/5*y^(-5)*x^(-2)-1/3*y^(-2)*x^(-3),y^(-8)*x^(-1)-7/5*y^(-6)*x^(-2)-1/3*y^(-3)*x^(-3)+7/15*y^(-1)*x^(-4)]]
```

ここで, 出力の $x^(-l)y^(-m)$ は $\left[\frac{1}{x^l y^m} \right]$ を意味する。

5 スタンダード基底とグレブナー基底

ここでは, 基底代数的局所コホモロジーの応用として, スタンダード基底計算アルゴリズム, 正規形の計算アルゴリズムとイデアルメンバーシップ問題の解法, グレブナー基底計算アルゴリズムを紹介する。

5.1 スタンダード基底

ここでは、基底代数的局所コホモロジーを用いたスタンダード基底計算アルゴリズムを紹介する。多項式 $f \in K[x]$ は、今までと同様、原点に孤立特異点を持つとし、形式冪級数環 $K[[x]]$ における f のヤコビデアル $\left\langle \frac{\partial f}{\partial x_1}, \dots, \frac{\partial f}{\partial x_n} \right\rangle$ を J で表す。

一般に H_f は、ベクトル空間として剰余空間 $K[[x]]/J$ の双対ベクトル空間と（微分形式 $dx_1 \wedge dx_2 \wedge \dots \wedge dx_n$ を固定することで）同一視できる。即ち、 H_f と $K[[x]]/J$ の間には、Grothendieck local residues が定める非退化な pairing

$$\text{res}_{\{O\}}(\cdot, \cdot) : H_f \times K[[x]]/J \longrightarrow K$$

が存在する。従って、形式冪級数環 $K[[x]]$ におけるヤコビデアル J は H_f により完全に特徴付けられることになる。

注意 Grothendieck local duality においては（微分形式 $dx_1 \wedge dx_2 \wedge \dots \wedge dx_n$ を固定した場合）、通常、 $\text{Ext}_{K[[x]]}^n(K[[x]]/J, K[[x]])$ を $K[[x]]/J$ の双対空間として定める。しかしこの通常の定式化では、pairing を与える $\text{res}_{\{O\}}(\cdot, \cdot)$ を計算することが実際には困難となってしまう。本稿では、代数的局所コホモロジー $H_{[0]}^n(K[x])$ の部分空間である H_f を $K[[x]]/J$ の双対空間として与え、 H_f の元は標準的な Leray 被覆による相対 Čech コホモロジーの要素として表現している。この表現法を採用していることで Grothendieck 留数が直ちに計算できる点に留意されたい。

さて、Grothendieck 留数を用いると、イデアル J は H_f により完全に特徴付けられることから、ベクトル空間 H_f の基底を利用することでイデアル J のスタンダード基底を計算できると考えるのは自然である。

実際、次の定理が示すように、第4章に与えたアルゴリズム **ALC** の出力を用いることで、スタンダード基底の構成が容易にできる ([30])。ただし、定理において今まで ξ と書いていたものを x と見なし、 ξ を x と表す。

定理 11 ([30]). リスト **SList** に属する p を $p = x^\tau + \sum_{\kappa < \tau} c_{(\tau, \kappa)} x^\kappa$ と書くとする。このとき、すべての **FList** の成分を次のように変換する。もし、 $\alpha \in \text{FList}$ で $\alpha \notin \text{LList}$ なら、多項式 x^α とし、もし、 $\alpha \in \text{FList}$ で $\alpha \in \text{LList}$ なら、多項式 $x^\alpha - \sum_{\kappa \in \text{TList}} c_{(\kappa, \alpha)} x^\kappa$ とする。この変換を **FList** に施すことで得られる多項式の集合は、イデアル J の局所全次数逆辞書式項順序に関するスタンダード基底となる。（もし、 $\kappa \in \text{TList}$ なら、**SList** に属する q が存在し、その形は $q = x^\kappa + \sum_{\beta < \kappa} c_{(\kappa, \beta)} x^\beta$ である。）

スタンダード基底を求める為には、項順序に注目し、スタンダード基底の計算に適するような H_f の基底を構成する必要があるが、第4章で述べた構成法は、スタンダード基底の計算に最も適した基底代数的局所コホモロジーを与える計算法といえる。スタンダード基底計算アルゴリズム ([30]) を以下に与える。

(アルゴリズム) **StandardBasis**(f)

Input: $f \in K[x]$,

Output: S : J の $K[[x]]$ におけるスタンダード基底

$S \leftarrow \{ \} ; (\text{MList}, \text{SList}, \text{TList}, \text{FList}) \leftarrow \text{ALC}(f)$

while $\text{FList} \neq []$ **do**

$\alpha \leftarrow \text{car}(\text{FList}) ; \text{FList} \leftarrow \text{cdr}(\text{FList}) ; s \leftarrow x^\alpha - \sum_{\kappa \in \text{TList}} c_{(\kappa, \alpha)} x^\kappa ; S \leftarrow S \cup \{s\}$

end-while

return(S)

例 12. 例 3 と同じ問題を考える。2 変数多項式 $f = x^3 + y^6 + x^2y^2 \in K[x, y]$ を見る。例 7 と 9 において, $\text{FList} = [(7, 0), (1, 3), (2, 0)]$, $\text{SList} = [p_1 = \xi\eta^2 - \frac{2}{3}\xi^2, p_2 = \eta^5 + \frac{9}{2}\xi\eta^3 - 3\xi^2\eta, p_3 = \eta^6 + \frac{9}{2}\xi\eta^4 - 3\xi^2\eta^2 + 2\xi^3]$, $\text{TList} = [(1, 2), (0, 5), (0, 6)]$ である。ここで, 図 8-4 のコーナー “*” は FList の成分であり, $K[[x, y]]$ において J のスタンダード基底の先頭項となる。(ξ は x , η は y に対応するものとする。)

- (1). FList から成分 $(0, 7)$ をとる。つまり, 今, 先頭項として y^7 を考える。多項式 p_1, p_2, p_3 は η^7 の項を持たないので, y^7 はそれ自体, スタンダード基底の成分となる。
 - (2). FList から成分 $(1, 3)$ をとる。つまり, 今, 先頭項として xy^3 を考える。このとき, p_2 が項 $\xi\eta^3$ を持ち, その係数は $\frac{9}{2}$ である。 $\text{ht}(p_2) = \eta^5$ より, $xy^3 - \frac{9}{2}xy^5$ を得る。
 - (3). FList から残りの成分 $(2, 0)$ をとる。頭項として x^2 を考える。このとき, p_1 が項 ξ^2 を持ち, その係数は $-\frac{2}{3}$ である。 $\text{ht}(p_1) = \xi\eta^2$ より, $x^2 + \frac{2}{3}xy^2$ を得る。
- したがって, J のスタンダード基底は $\left\{ y^7, xy^3 - \frac{9}{2}xy^5, x^2 + \frac{2}{3}xy^2 \right\}$ である。項順序は局所全次数逆辞書項順序 $(1 \succ y \succ x \succ y^2 \succ xy \succ x^2 \succ y^2 \succ \dots)$ である。

5.2 正規形計算アルゴリズムとメンバーシップ問題

ここでは, まず, Grothendieck 留数を用いることでイデアルメンバーシップ問題が解けることを示し, 更に, 正規形計算を行えることを紹介する。

さて, H_f の基底 $\psi_1, \psi_2, \dots, \psi_\mu$ (実際にはこれらの多項式表現) が与えられているとする。ここでいま, 多項式 $h(x) \in K[x]$ (もしくは形式的冪級数 $h(x) \in K[[x]]$) が与えられたとする。この時, $h(x)$ がヤコビイデアル J に属すことは, Grothendieck 双対定理より,

$$\text{res}_{\{0\}}(h, \psi_i) = 0, \quad i = 1, 2, \dots, \mu$$

と同値であることが従う。即ち, 基底 $\psi_1, \psi_2, \dots, \psi_\mu$ を用いることでイデアルメンバーシップ問題が完全に解けることが分かる。ここまでの議論では, 代数的局所コホモロジー類 $\psi_1, \psi_2, \dots, \psi_\mu$ は H_f の基底でありさえすればよく, 特別な性質を持つような必要はないことに注意しよう。第 4 章のアルゴリズムにより得る基底代数的局所コホモロジー類は, 単にベクトル空間 H_f の基底であるというのではなく, 様々な計算に適した「良い基底」である。実際, **ALC** の出力を用いると一般の基底を使用した場合に比べ, 計算効率よくイデアルメンバーシップ問題を解くことができることを注意しておきたい。ここではイデアルメンバーシップ問題に関する具体的説明は省き, 以下, 正規形計算アルゴリズムについて述べることにする。

まず, あらためて記号を整理しておく。多項式 f の $K[[x]]$ におけるヤコビイデアルを J とする。与えられた多項式 h に対し J を法とする h の正規形を計算する方法を与える。もちろんこの方法により, J のメンバーシップ問題を解くことができる。

まず, 2 つの集合 ML, DL を

$$\text{ML} = \{\text{mdeg}(g) | g \in \text{MList}\}, \quad \text{DL} = \text{ML} \cup \text{set}(\text{TList}) \cup \text{set}(\text{LList})$$

により定義する。定理 11 から次の 2 つの系が従う。

系 13. $\text{MList}, \text{TList}, \text{LList}$ を **ALC**(f) の出力とする。もし, $\lambda \notin \text{DL}$ ならば, そのとき, $x^\lambda \in J$ である。

系 14. ここで任意の $p \in \text{SList}$ を $p = x^\tau + \sum_{\kappa \prec \tau} c_{(\tau, \kappa)} x^\kappa$ と書くようにする。このとき, 任意の $\alpha \in \text{set}(\text{TList}) \cup \text{set}(\text{LList})$ に対して, $x^\alpha \equiv \sum_{\kappa \in \text{TList}} c_{(\kappa, \alpha)} x^\kappa \pmod{J}$ となる。

((注) もし $\kappa \in \text{TList}$ であれば, SList に q が存在し形として $q = x^\kappa + \sum_{\beta \prec \kappa} c_{(\kappa, \beta)} x^\beta$ と書くことができる。)

この2つの系と定理 11 に基づくことで、正規形の計算アルゴリズムを導出することができる。

(アルゴリズム) Normal-form(f, h)

Input: f, h : $K[x]$ の多項式,

Output: p : $J(f$ のヤコビデアル) を法とする h の正規形 (i.e., $p \equiv h \pmod{J}$).

(0) (MList, SList, TList, FList, LList) \leftarrow ALC(f)

ML $\leftarrow \{\text{mdeg}(g) | g \in \text{MList}\}$; DL $\leftarrow \text{ML} \cup \text{set}(\text{TList}) \cup \text{set}(\text{LList})$

(1) $h(x)$ を $h(x) = \sum_{\lambda \in \text{DL}} h_{\lambda} x^{\lambda} + \sum_{\lambda' \notin \text{DL}} h_{\lambda'} x^{\lambda'}$ と分解する。系 13 より, $h(x) \equiv \sum_{\lambda \in \text{DL}} h_{\lambda} x^{\lambda} \pmod{J}$ 。

(2) $\sum_{\lambda \in \text{DL}} h_{\lambda} x^{\lambda} = \sum_{\tau \in \text{ML}} h_{\tau} x^{\tau} + \sum_{\gamma \in \text{DL} \setminus \text{ML}} h_{\gamma} x^{\gamma}$ と分解する。

(3) 系 14 より, $\forall \alpha \in \text{set}(\text{TList}) \cup \text{set}(\text{LList}), x^{\alpha} \equiv \sum_{\kappa \in \text{TList}} c_{(\kappa, \alpha)} x^{\kappa} \pmod{J}$ 。

第 2 パート $\sum_{\gamma \in \text{DL} \setminus \text{ML}} h_{\gamma} x^{\gamma}$ を $\sum_{\tau \in \text{TList}, \gamma \in \text{DL} \setminus \text{ML}} c_{(\gamma, \tau)} x^{\tau}$ に変換する。

(4) ここで, $p = \sum_{\tau \in \text{ML}} h_{\tau} x^{\tau} + \sum_{\tau \in \text{TList}, \gamma \in \text{DL} \setminus \text{ML}} c_{(\gamma, \tau)} x^{\tau}$ を得る。これは $p \equiv h \pmod{J}$ となる。

return(p)

5.3 グレブナー基底

ここでは、基底代数的局所コホモロジーを用いたグレブナー基底計算アルゴリズムを紹介する。 $I = \left\langle \frac{\partial f}{\partial x_1}, \dots, \frac{\partial f}{\partial x_n} \right\rangle$ を多項式環 $K[x]$ において $\frac{\partial f}{\partial x_1}, \dots, \frac{\partial f}{\partial x_n}$ が生成するイデアルとし、 I の準素イデアル分解の成分であり、原点における極大イデアル $\langle x_1, x_2, \dots, x_n \rangle$ を素イデアルとする準素イデアルを $J \subset K[x]$ とおく。このヤコビデアル J のグレブナー基底を求める方法を与える。この計算法はアルゴリズム StandardBasis を広域 (global) 項順序 (1 が最小な項順序) にも適応するように変えたものである。

(アルゴリズム) GröbnerBasis(f)

Input: $f \in K[x]$: 多項式, \prec : 広域 (global) 項順序,

Output: S : \prec に関しての J のグレブナー基底。

(0). $S \leftarrow \{ \}$

(MList, SList, TList, LList, FList) \leftarrow ALC(f)

$[\alpha_1, \dots, \alpha_s] \leftarrow$ 項順序 \prec に関して append(TList, LList) の成分を小さい方から並べる。

(すなわち, $\xi^{\alpha_1} \prec \dots \prec \xi^{\alpha_s}$)

(1). $[p_1, \dots, p_m] = \text{SList}$ とする。ここで SList の係数行列 Φ を作成する。

$$\text{I.e., } \begin{pmatrix} p_1 \\ \vdots \\ p_m \end{pmatrix} = \Phi \begin{pmatrix} \xi^{\alpha_1} \\ \vdots \\ \xi^{\alpha_s} \end{pmatrix}.$$

$\Phi' \leftarrow$ 標準変形を施し Φ の階段行列を計算する。

$$(2). \begin{pmatrix} q_1 \\ \vdots \\ q_m \end{pmatrix} \leftarrow \Phi' \begin{pmatrix} \xi^{\alpha_1} \\ \vdots \\ \xi^{\alpha_s} \end{pmatrix}$$

$$TL \leftarrow \{mdeg(q_1), \dots, mdeg(q_m)\}$$

$$ML \leftarrow \{mdeg(g) | g \in MList\}$$

(3). $G \leftarrow \langle \xi^\gamma | \forall \gamma \in nb(TL \cup ML) \setminus (TL \cup ML) \rangle$ の簡約グレブナー基底を計算する。

$$KList \leftarrow list(\{mdeg(t) \in \mathbb{N}^n | \forall t \in G\}) \quad /* \text{ FList に相当するもの。図 15 での “*”}。 */$$

(4). **while** $KList \neq []$ **do**

$$\sigma \leftarrow car(KList)$$

$$KList \leftarrow cdr(KList)$$

$$s \leftarrow x^\sigma - \sum_{\lambda \in TL} c_{(\lambda, \sigma)} x^\lambda \quad /* \text{ もし } \lambda \in TL \text{ なら, } \exists q = x^\lambda + \sum_{\beta \prec \lambda} c_{(\lambda, \beta)} x^\beta \in \{q_1, \dots, q_m\} */$$

$$S \leftarrow S \cup \{s\}$$

end-while

$$\text{return}(S)$$

(3) において、定理 11 で重要であった FList に相当するものを計算している。定理 11 より、これらの元はグレブナー基底で先頭項になる。このアルゴリズムは準素イデアル分解を介さずに、準素イデアル J のグレブナー基底を計算していることに注意されたい。

例 15. 例 3 と同じ問題を考える。2 変数多項式 $f = x^3 + y^6 + x^2 y^2 \in K[x, y]$ を見る。広域項順序として全次数辞書式項順序 \prec 、ただし $y \prec x$ を考える。 f の準素イデアル分解の成分で原点に台を持つものを J とおき、この準素イデアル J のグレブナー基底を考える。例 7 と 9 より、 $SList = [p_1 = \xi \eta^2 - \frac{2}{3} \xi^2, p_2 = \eta^5 + \frac{9}{2} \xi \eta^3 - 3 \xi^2 \eta, p_3 = \eta^6 + \frac{9}{2} \xi \eta^4 - 3 \xi^2 \eta^2 + 2 \xi^3]$, $TList = [(1, 2), (0, 5), (0, 6)]$, $LList = [(0, 6), (1, 4), (0, 5), (2, 2), (1, 3), (3, 0), (2, 1), (1, 2), (2, 0)]$ である。(ξ は x , η は y に対応するものとする。)

まず、項順序に関して $\text{append}(TList, LList)$ の成分を小さいものから並べる。このとき次のようになる。

$$(2, 0), (1, 2), (2, 1), (3, 0), (1, 3), (2, 2), (0, 5), (1, 4), (0, 6).$$

次に $SList$ の係数行列 Φ を作成する。

$$\Phi = \begin{pmatrix} \xi^2 & \xi \eta^2 & \xi^2 \eta & \xi^3 & \xi \eta^3 & \xi^2 \eta^2 & \eta^5 & \xi \eta^4 & \eta^6 \\ -\frac{2}{3} & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & -3 & 0 & \frac{9}{2} & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & -2 & 0 & -3 & 0 & \frac{9}{2} & 1 \end{pmatrix}$$

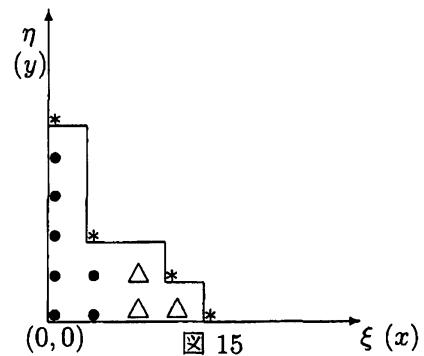
Φ の階段行列は次の行列 Φ' である。

$$\Phi' = \begin{pmatrix} \xi^2 & \xi \eta^2 & \xi^2 \eta & \xi^3 & \xi \eta^3 & \xi^2 \eta^2 & \eta^5 & \xi \eta^4 & \eta^6 \\ 1 & -\frac{3}{2} & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & -\frac{3}{2} & 0 & -\frac{1}{3} & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & \frac{3}{2} & 0 & -\frac{9}{4} & -\frac{1}{2} \end{pmatrix}$$

今、 $ML = \{(0, 0), (1, 0), (0, 1), (1, 1), (0, 2), (0, 3), (0, 4)\}$, $TL = \{(2, 0), (3, 0), (2, 1)\}$ である。図 15 において、 \bullet は ML の成分を意味し、 \triangle は TL の成分を意味する。イデアル $\langle \xi^\gamma | \forall \gamma \in nb(TL \cup ML) \setminus (TL \cup ML) \rangle$ の簡約グレブナー基底 G を計算すると $G = \{\eta^5, \xi \eta^2, \xi^3 \eta, \xi^4\}$ となる。すなわち、図のコーナーが $KList = [(0, 5), (1, 2), (3, 1), (4, 0)]$ である。図 15 において、“*” は $KList$ の成分を意味する。

ここで、系 14 より、次の関係を得る。

$$\begin{cases} xy^2 \equiv -\frac{3}{2}x^2, \\ xy^3 \equiv -\frac{3}{2}x^2y, \\ x^2y^2 \equiv \frac{3}{2}x^3, \quad \text{mod } J \\ y^5 \equiv -\frac{1}{3}x^2y, \\ xy^4 \equiv -\frac{9}{4}x^3, \\ y^6 \equiv -\frac{1}{2}x^3. \end{cases}$$



今、アルゴリズム StandardBasis の流れと同様な方法で J のグレブナー基底を求めることができる。即ち、定理 11 と同様の変換を施すことで、 J のグレブナー基底

$$\left\{ y^5 + \frac{1}{3}x^2y, x^4, x^3y, xy^2 + \frac{3}{2}x^2 \right\}$$

を得る。

我々はグレブナー基底計算アルゴリズムを計算機代数システム Risa/Asir に実装した。ここでは、我々の実装がどのように出力するか示す。

例 16. 多項式 $f = x^4 + xy^7 + x^2y^5$ によって定義される W_{25} 特異点を考える。 J の広域全次数辞書式項順序（ただし、 $x \succ y$ ）でのグレブナー基底は次となる。

$[x^6, yx^5, 5/7x^5y^2x^4, -196/45yx^4y^4x^3, 28/9x^4y^5x^2, 5/7y^4x^2y^6x, 4x^3+2y^5x+y^7]$

6 パラメータ付き基底代数的局所コホモロジーとスタンダード基底

ここでは、定義多項式 f が係数にパラメータを含んでいる場合を考える。一般に、イデアルがパラメータを含む場合、そのイデアルの性質・構造はパラメータの値に依存する。もちろん、スタンダード基底も同様に、パラメータ値に依存し変化する。従って、パラメータ付きスタンダード基底計算では、パラメータの値による場合分けが必要である。すなわち、パラメータ空間を適切に分割 (stratify) し更に各 stratum 上でスタンダード基底の計算を行う必要がある。

ここでは、前章で与えた基底代数的局所コホモロジー計算アルゴリズム ALC を入力がパラメータ付き定義多項式である場合に対応すべく拡張する。

アルゴリズム ALC 内でパラメータの値により計算を分枝させる必要が生じる箇所は 3 箇所ある。『ヤコビイデアルの次元』、『単項のグレブナー基底計算 STEP1』、『連立方程式 Head.main (3)』である。そこで、この 3 箇所について考える。

(1) ヤコビイデアルの次元

アルゴリズム ALC はヤコビイデアル $\langle \frac{\partial f}{\partial x_1}, \dots, \frac{\partial f}{\partial x_n} \rangle$ が特異点 O において零次元であることを仮定している。入力する多項式がパラメータを持つ場合そのヤコビイデアルはパラメータの値によって次元が変化

する可能性がある。スタンダード基底計算に基底代数的局所コホモロジーを使うためには、計算の停止性も含めてイデアルが零次元である確証が必要である。さて、どのように零次元かどうかを判定するか？

計算停止性を含めた局所的なイデアルの次元分類方法は知られていない。そこで、我々はヤコビイデアルの広域的な次元を考え、広域的に（多項式環のイデアルとして）次元が零次元か零次元でないかでパラメータ空間を分割する。この方法はヤコビイデアルの包括的グレブナー基底系 (Comprehensive Gröbner Systems) [20, 24, 31] を計算することで、パラメータ空間が分割されるので、各 stratum 上でイデアルの次元を判定（例えば [6] 第 9 章）すれば可能である。ここで、広域的に零次元であれば局所的にも零次元であるので広域的な次元が零である strata に対してのみ、まずは計算を実行するようにする。

この方法での問題点は、広域的に零次元でなくても局所的には零次元になるものが存在するという点である。この場合、広域的に零次元であるものに対する計算が終了した後に何らかの計算をする必要がある。現実的な対処の仕方として、例えば、広域的には零次元でないと判定された stratum 上で基底代数的局所コホモロジー計算を行い、基底の元の個数が予め指定した数値を超えた段階で強制的に計算を止めるという方法が考えられる。数値の指定に際しては、定義多項式から何らかの方法で H_f の次元の上限を見積もり、その値を強制終了を行う数値として採用することになる。

(2) 単項のグレブナー基底

アルゴリズム STEP 1 (2) において、 $\frac{\partial f}{\partial x_1}, \dots, \frac{\partial f}{\partial x_n}$ を構成する単項のグレブナー基底を計算する必要がある。パラメータの値によってグレブナー基底は変化するので、この箇所においても包括的グレブナー基底系を計算する必要がある。包括的グレブナー基底系の出力はパラメータ空間が分割された形 (stratum) とそれ上のグレブナー基底を出力する。各 stratum 上で基底代数的局所コホモロジー計算を行うが、もちろんヤコビイデアルが零次元の時である。

(3) 連立方程式

Head_main (3) において、連立方程式を解く必要がある。係数がパラメータを含むような線型連立方程式はパラメータの値によって解の構造が変化する。線型な連立方程式より、ガウスの消去法を係数が零になるか零にならないかで計算を分岐させることにより、パラメータ付き線型連立方程式の解法を構成できる。もちろん、この場合も各 stratum 上で分枝する。

この 3 箇所の改良を ALC に加えることによって、パラメータ付き基底代数的局所コホモロジー計算は可能である。次元の分類、単項のグレブナー基底計算以外は、基本的には線型代数の手法のみを用いているので計算の流れは ALC と同じである。以下は計算の概略である。

(アルゴリズム) parametricALC(f)

Input: $f \in K[x, a]$ ただし、 f はパラメータ $a = (a_1, \dots, a_m)$ を持つ変数 x についての多項式、

Output: $\mathcal{R} = \{R_1, \dots, R_k\}$, $\mathcal{G} = \{(S_1, H_1, F_1), \dots, (S_l, H_l, F_l)\}$,

ここで $S_i \subset \mathbb{A}^m$ stratum, H_i は S_i 上での f の基底代数的局所コホモロジー, F_i は S_i 上で基底代数的局所コホモロジーとなることが出来なかった主項の候補の集合 (FList), \mathcal{R} は零次元とならない stratum R_j の集合, $i = 1, \dots, k$, $j = 1, \dots, l$. また、 $R_1 \cup \dots \cup R_k \cup S_1 \cup \dots \cup S_l = \mathbb{A}^m$ である。

初期化: $I \leftarrow \langle \frac{\partial f}{\partial x_1}, \dots, \frac{\partial f}{\partial x_n} \rangle$

1. I の次元の分類

$\mathcal{R} \leftarrow$ 零次元とならない strata 達; $\mathcal{Z} \leftarrow$ 零次元となる strata 達

2. $\mathcal{T} = \{(T_1, G_1), \dots, (T_d, G_d)\} \leftarrow \frac{\partial f}{\partial x_1}, \dots, \frac{\partial f}{\partial x_n}$ を構成する単項から生成される包括的グレブナー基底系の計算。(T_1, \dots, T_s は strata, G_i は T_i 上でグレブナー基底となる。)

$\mathcal{L} \leftarrow \{(S_i, G_i) | S = T_i \cap Z \neq \emptyset, (T_i, G_i) \in \mathcal{T}, i = 1, \dots, s\}$ /*零次元の stratum との共通部分*/

3. \mathcal{L} の各要素の stratum 上で線型結合の形の代数的局所コホモロジーの計算

(注意) パラメータ付き線型連立方程式を解く必要がある。また、連立方程式に解が存在するか、存在しないかで分枝する可能性がある。

すべての stratum 上で停止条件 (Head_main (*2)) を満たすまで計算する。

アルゴリズム **parametricALC** により、ヤコビデアルが広域的に零次元になる strata 達上でパラメータ付き基底代数的局所コホモロジーの計算が可能である。この計算後、零次元とならない strata 達 \mathcal{R} の考察が必要である。例えば、ある数字を入力し、各 stratum 上で `append(MList, SList)` の要素数とその数を超えたら強制的に計算を停止しその strata 達を返し、もし、停止条件まで計算できればその strata と、MList, SList を返すようにすればよい。これによりヤコビデアルが零次元の基底代数的局所コホモロジーを計算することができる。

この計算方法により得られた基底代数的局所コホモロジーと主項にならなかった集合 F_i によりヤコビデアル J のスタンダード基底は、各 stratum に定理 11 を適用することで容易に求まる。

我々はこのアルゴリズムを計算機代数システム Risa/Asir [23] に実装した。この章を締めくくるにあたり。簡単な例を取り上げ我々の実装の出力を紹介する。以後、 $\mathbb{V}(g_1, \dots, g_s)$ は $g_1, \dots, g_s \in K[a]$ のアフィン多様体 $\{(a_1, \dots, a_m) \in \mathbb{A}^m | g_i(a_1, \dots, a_m) = 0 \text{ for all } 1 \leq i \leq s\}$ を意味する。

原点に特異点を持つ多項式 $f = x^3 + axy^3 + by^4 + xy^4$ を考える (a, b はパラメータである)。我々が実装したプログラム `p_co_std` に、多項式 f 、パラメータのリスト $[a, b]$ 、変数のリスト $[x, y]$ 、 $1 \succ y \succ x \succ y^2 \succ yx \succ x^2 \succ \dots$ なる局所全次数辞書式項順序を表す引数 1 を与えると、`p_co_std` は、各 stratum において、パラメータ付き基底代数的局所コホモロジーとスタンダード基底を次のように出力する。

```
[391] p_co_std(x^3+a*x*y^3+b*y^4+x*y^4,[a,b],[x,y],1,1);
```

```
non zero-dim.
```

```
[]
```

```
Bases of parametric cohomology & standard bases
```

```
[[a,b],[1]]
```

```
[y^(-5)*x^(-1)-1/3*y^(-1)*x^(-3),y^(-6)*x^(-1)-1/3*y^(-2)*x^(-3),y^(-7)*x^(-1)-1/3*y^(-3)*x^(-3)] [y^(-1)*x^(-1),y^(-2)*x^(-1),y^(-1)*x^(-2),y^(-3)*x^(-1),y^(-2)*x^(-2),y^(-4)*x^(-1),y^(-3)*x^(-2)]
```

```
No. of coho. is 3+7
```

```
Standard basis
```

```
[x^2+1/3*y^4,y^3*x,y^7]
```

```
[[a],[a,b]]
```

```
[] [y^(-1)*x^(-1),y^(-2)*x^(-1),y^(-1)*x^(-2),y^(-3)*x^(-1),y^(-2)*x^(-2),y^(-3)*x^(-2)]
```

```
No. of coho. is 0+6
```

```
Standard basis
```

```
[x^2,y^3]
```

```
[[b],[a,b]]
```

```
[y^(-4)*x^(-1)-1/3*a*y^(-1)*x^(-3),y^(-5)*x^(-1)+(-1/3*y^(-1)-1/3*a*y^(-2))*x^(-3)]
```

```
[y^(-1)*x^(-1),y^(-2)*x^(-1),y^(-1)*x^(-2),y^(-3)*x^(-1),y^(-2)*x^(-2)]
```

No. of coho. is 2+5

Standard basis

```
[x^2+1/3*y^4+1/3*a*y^3,y^2*x,y^5]
```

```
[[0],[-b*a]]
```

```
[-3/4*a*y^(-4)*x^(-1)+b*y^(-3)*x^(-2)+1/4*a^2*y^(-1)*x^(-3)][y^(-1)*x^(-1),y^(-2)*x^(-1),y^(-1)*x^(-2),y^(-3)*x^(-1),y^(-2)*x^(-2)]
```

No. of coho. is 1+5

Standard basis

```
[b*x^2-1/4*a^2*y^2*x,3/4*a*y^2*x+b*y^3]
```

プログラムは、まず最初に零次元でないパラメータ値のなす strata を出力するように作られている。この例の場合は、すべてのパラメータに対し零次元であるので、空リスト [] を返している。次に各 stratum とそこでの基底代数的局所コホモロジーとその個数、スタンダード基底を返す。

- パラメータが $\mathbb{V}(a, b)$ に属する場合、すなわち $a = b = 0$ のとき、基底代数的局所コホモロジーとして 10 個の元を返し、スタンダード基底は $\{x^2 + \frac{1}{3}y^4, y^3x, y^7\}$ 。
- パラメータが $\mathbb{V}(a) \setminus \mathbb{V}(a, b)$ に属する場合、基底代数的局所コホモロジーは 6 個の元を返し、スタンダード基底は $\{x^2, y^3\}$ 。
- パラメータが $\mathbb{V}(b) \setminus \mathbb{V}(a, b)$ に属する場合、基底代数的局所コホモロジーは 7 個の元を返し、スタンダード基底は $\{x^2 + \frac{1}{3}y^4 + \frac{1}{3}ay^3, y^2x, y^5\}$ 。
- パラメータが $\mathbb{A}^2 \setminus \mathbb{V}(ab)$ に属する場合、基底代数的局所コホモロジーは 6 個の元を返し、スタンダード基底は $\{bx^2 - \frac{1}{4}a^2y^2x, \frac{3}{4}ay^2x + by^3\}$ 。

この例が示すように、実装したプログラムはパラメータ空間内の stratification の各 stratum とそれに対応する基底代数的局所コホモロジーとスタンダード基底の組を自動的に出力する。

紙面の関係上、簡単な例を扱った。この例ではあまり複雑な分枝が現れていないが、一般には分枝が多く現れる。分枝が多くなると stratification の計算自体、計算機に頼らざるを得ない。この実装により、複雑な計算を要する特異点の研究が計算機を介して行うことができるようになったことは意義あることである。

7 最後に

本稿で紹介したアルゴリズムはすべてフリーの計算機代数システム Risa/Asir 上で実装されている。2011 年度中には第一著者のホームページ上で公開する予定である。

謝辞

本研究において第一著者は科学研究費 (課題番号:22740065)、第二著者は科学研究費 (課題番号:21740108)、第三著者は科学研究費 (課題番号:70155076) の助成を受けている。

参考文献

- [1] 阿部隆行, 田島慎一, 孤立特異点に付随する代数的局所コホモロジーとヤコビイデアルに対するグレブナー基底の計算法, 数理解析研究所講究録 **1514**, pp.141 – 147, (2006).
- [2] M. E. Alonso, S.M. Marinari, and T. Mora, The big mother of all dualities: Möller algorithm, Comm. in Algebra **31**, pp. 783 – 818, (2003).
- [3] C. Bănică and O. Stănăşilă, Méthodes Algébriques dans la Théorie Globale des Espaces Complexes, Gauthier-Villars, (1974).
- [4] M. P. Brodmann and R. Y. Sharp, Local Cohomology, Cambridge Univ. Press (1998).
- [5] B. Buchberger, Ein Algorithmus zum Auffinden der Basiselemente des Restklassenrings nach einem nulldimensionalen Polynomideal, Ph.D. Thesis, Universität Innsbruck, Austria, (1965).
- [6] D. Cox, J. Little, and D. O’Shea, Ideals, Varieties, and Algorithms, Springer, (1992).
- [7] H. -G. Gräbe, The tangent cone algorithm and homogenization, J. Pure Appl. Algebra **97**, pp. 303 – 312, (1994).
- [8] H. Grassmann, G.-M. Greuel, B. Martin, W. Neumann, G. Pfister, W. Pohl, H. Schönemann and T. Siebert, On an implementation of standard bases and syzygies in SINGULAR, AAECC **7**, pp. 235 – 249, (1996).
- [9] A. Grothendieck, Théorème de dualité pour les faisceaux algébriques cohérents, Séminaire Bourbaki **149**, Paris, (1957).
- [10] A. Grothendieck, Local Cohomology, notes by R. Hartshorne, Lecture Notes in Math. **41**, Springer, (1967).
- [11] H. Hironaka, Resolution of singularities of an algebraic variety over a field of characteristic zero, Ann. Math. **79**, pp. 109 – 326, (1964).
- [12] S. B. Iyengar, G. J. Leuschke, A. Leykin, C. Miller, E. Miller, A. K. Singh and U. Walther, Twenty-Four Hours of Local Cohomology, AMS, (2007).
- [13] D. Lazard, Gröbner bases, Gaussian elimination, and resolution of systems of algebraic equations, Lecture Note in Compu. Sci. **162**, pp. 146 – 156, (1983).
- [14] G. Lyubeznik, Local Cohomology and its Applications, Dekker, (2002).
- [15] M.G. Marinari, H.M. Möller and T. Mora, Gröbner bases of ideals given by dual bases, Proc. International Symposium on Symbolic and Algebraic Computation, pp. 55 – 63, AMC-press, (1991).
- [16] T. Mora, An algorithm to compute the equations of tangent cones, Lecture Notes in Compu. Sci. **144**, pp. 158 – 165, Springer, (1982).
- [17] T. Mora, G. Pfister and C. Traverso, An introduction to the tangent cone algorithm, Adv. in Computing Research, issued in robotics and nonlinear geometry **6**, pp. 199 – 270, (1992).
- [18] B. Mourrain, Isolated points, duality and residues, Journal of Pure and Applied Algebra, **117** & **118**, pp. 469 – 493, (1997).
- [19] 鍋島克輔, 中村弥生, 田島慎一, パラメータ付き零次元代数的局所コホモロジーを用いたパラメトリック・スタンダード基底計算について, 京都大学数理解析研究所講究録掲載予定

- [20] K. Nabeshima, A speed-up of the algorithm for computing comprehensive Gröbner systems, Proc. ISSAC'07, pp. 299 – 306. AMC-Press (2007).
- [21] Y. Nakamura and S Tajima, Unimodal singularities and differential operators, Sémin. Congr. **10**, pp. 191 – 208, Soc. Math. France, (2005).
- [22] Y. Nakamura and S Tajima, On weighted-degrees for algebraic local cohomologies associated with semiquasihomogeneous singularities, Advanced Studies in Pure Mathematics, **46**, pp. 105 – 117, (2007).
- [23] M. Noro and T. Takeshima, Risa/Asir- A Computer Algebra System, Proc. ISSAC'92, pp.387 – 396, ACM-Press, (1992).
- [24] A. Suzuki and Y. Sato, A simple algorithm to compute comprehensive Gröbner bases using Gröbner Bases, Proc. ISSAC'06, pp. 326 – 331. AMC-Press (2006).
- [25] 田島慎一, 零次元代数的局所コホモロジーの計算法とスタンダード基底計算について, 数理解析研究所講究録 **1456**, pp.126 – 132, (2005).
- [26] 田島慎一, 零次元代数的局所コホモロジーの計算法とスタンダード基底計算について II, 数理解析研究所講究録 **1568**, pp.74 – 80, (2007).
- [27] S. Tajima and Y. Nakamura, Algebraic local cohomology class attached to quasi-homogeneous isolated hypersurface singularities. Publ. Res. Inst. Math. Sci. **41**, pp.1 – 10, (2005)
- [28] S. Tajima and Y. Nakamura, Computational aspects of Grothendieck local residues. Sémin. Congr. **10**, Soc. Math. France, pp.287 – 305, (2005).
- [29] S. Tajima and Y. Nakamura, Annihilating ideals for an algebraic local cohomology class. Journal of Symbolic Computation **44**, pp.435 – 448, (2009) .
- [30] S. Tajima, Y. Nakamura and K. Nabeshima, Standard bases and algebraic local cohomology for zero dimensional ideals, Advanced Studies in Pure Mathematics **56**, pp. 341 – 361, (2009).
- [31] V. Weispfenning, Comprehensive Gröbner bases, Journal of Symbolic Computation **14**, pp. 1 – 29 (1992).